

FINAL DE ALGORITMOS Y ESTRUCTURAS DE DATOS II
21/04/2021

PARA CONSIDERARSE APROBADO DEBEN
ESTAR BIEN AL MENOS 3 EJERCICIOS

1) (2 pts.) Se desea modelar una ventanilla de un banco donde los clientes son atendidos ni bien se termina con el anterior. Identificar cuál es el problema (si lo hubiese) y explicar por qué lo es en cada uno de los siguientes casos.

a) generadores

inaugurar: -> banco

llega_cliente: banco x cliente -> banco

atender_cliente: banco b -> banco (largo_de_la cola(b)>0)

pase_el_que_sigue: banco b -> banco (largo_de_la cola(b)>0)

observadores

largo_de_la cola: banco -> nat

b) generadores

inaugurar: -> banco

llega_cliente: banco x cliente -> banco

observadores

largo_de_la cola: banco -> nat

otras operaciones

atender_cliente: banco b -> banco (largo_de_la cola(b)>0)

c) generadores

inaugurar: -> banco

llega_cliente: banco x cliente -> banco

observadores

largo_de_la cola: banco -> nat

otras operaciones

atender_cliente: banco b -> banco (largo_de_la cola(b)>0)

clientes_atendidos: banco -> secu(cliente)

2) (2 pts.) El diseño de un módulo requiere la utilización de un conjunto y un diccionario con complejidad a lo sumo logarítmica para todas sus operaciones. ¿Alguna de las siguientes alternativas de diseño le parece más apropiada que las demás? Justifique. Si no, proponga una usted, justificando.

a) conjunto sobre diccionario sobre AVL.

b) conjunto sobre AVL, diccionario sobre AVL, AVL sobre punteros.

c) AVL sobre ABB sobre AB sobre punteros, conjunto sobre AVL, diccionario sobre AVL.

d) Conjunto sobre AVL, diccionario sobre tabla de hashing abierta con conjuntos en las celdas.

3) (2pts.) Determine si las siguientes afirmaciones son verdaderas o falsas, justificando.

- a) El invariante vale en la pre pero no en la post de todas las funciones de la interfaz.
- b) El invariante vale en la pre y en la post de todas las funciones de implementación que se corresponden con funciones de la interfaz (ejemplo: iAgregar).
- c) El invariante vale en la pre y en la post de todas las funciones auxiliares de la implementación.
- d) La función de abstracción provee una explicación de cómo se usa la estructura de representación.
- e) Por ende, determina la complejidad de las operaciones.

4) (2pts.) Se deben ordenar los parciales de 1000 alumnos, en base a su apellido. Para facilitar la práctica, los m docentes deciden dividir la tarea entre ellos. Recomiende dos posibles métodos para que utilicen (los métodos propuestos pueden o no corresponderse con algoritmos conocidos o con combinaciones de ellos).

Tenga en cuenta:

- Debe ser realizable por personas.
- Por más de una.
- Se pretende además que sea eficiente.

Calcule la complejidad asintótica de los métodos propuestos. Justifique sus respuestas.

5) (2pts.)

- a) Escribir un algoritmo que calcule, dado un ABB no vacío con claves enteras, con costo computacional lineal, el valor mínimo de las claves memorizadas en cada subárbol. Más precisamente, el algoritmo debe producir, como salida, la secuencia de parejas $(k, \min(k))$, con $\min(k)$ el mínimo valor del subárbol que contiene k en la raíz. El orden relativo de las parejas en la secuencia no es relevante.
- b) ¿Qué puede decir de la complejidad de su algoritmo en el caso de que se aplicara a un árbol AVL?